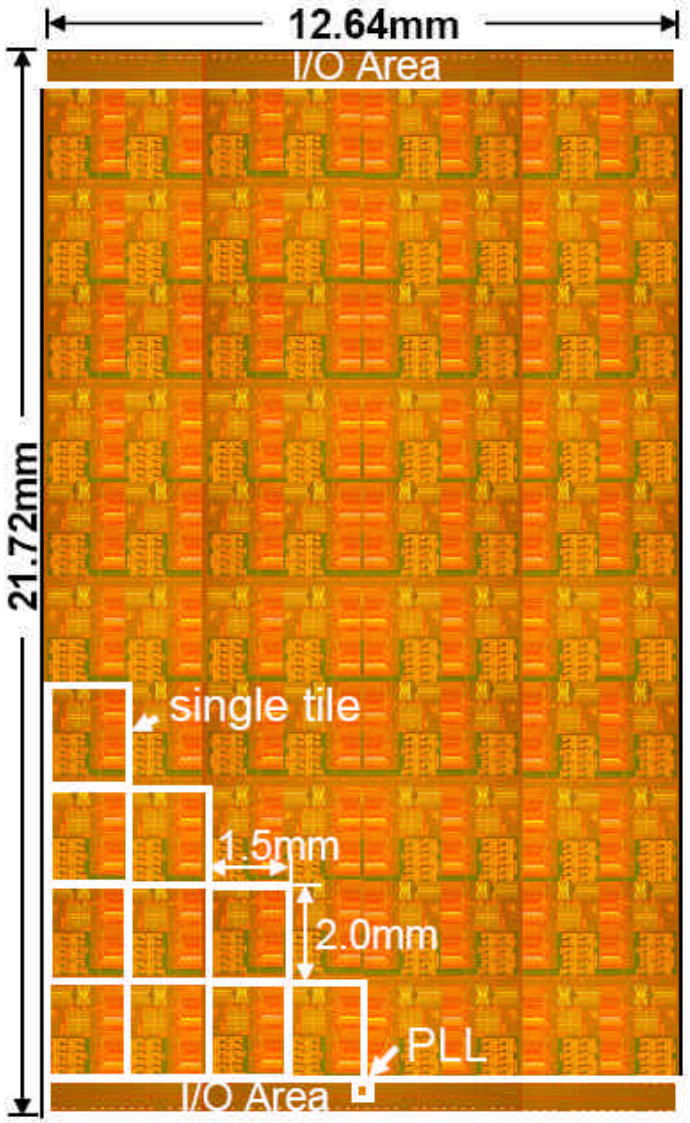
Design Technologies for Networks-on-Chip

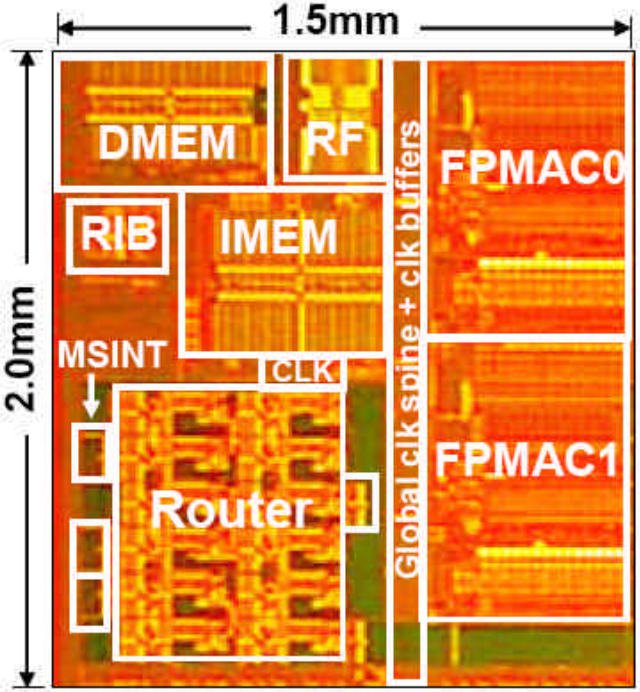
Giovanni De Micheli

EPFL

Federico Angiolini, Srinivasan Murali,
Luca Benini, David Atienza, Antonio Pullini

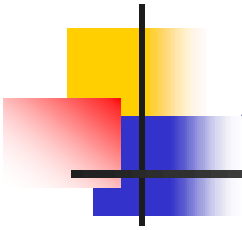
Intel's 80-tile NoC





Vangal et al. ISSCC 2007

Technology	65nm CMOS Process
Interconnect	1 poly, 8 metal (Cu)
Transistors	100 Million
Die Area	275 mm^2
Tile Area	3 mm^2
Package	1248 pin LGA, 14 layers, 343 signal pins

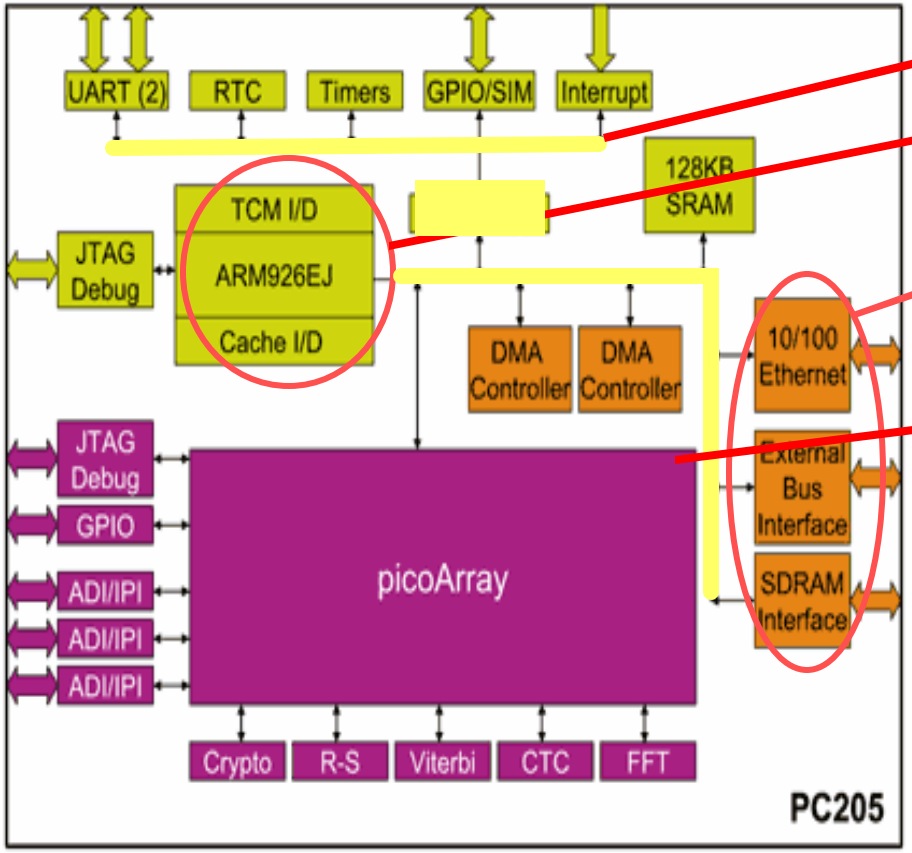


Application domains

- Multiprocessors on chip
  - Homogenous fabric
  - Designed for performance
  - General purpose
- Application-specific SoCs
  - Heterogeneous structure
  - QoS and power constraints
  - Domain specific software

Embedded SoC Trend

Heterogeneous clusters

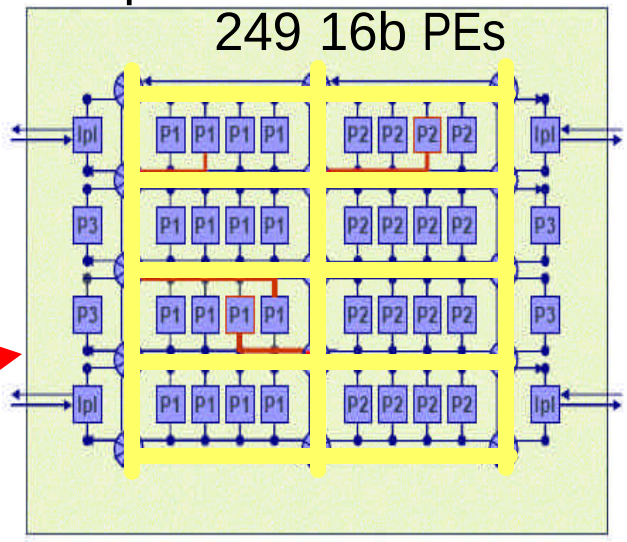


Wireless Networks Mesh nodes, Picocells

Multi-hop interconnect

GP core

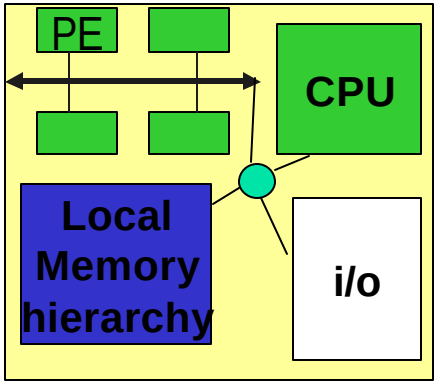
IOs

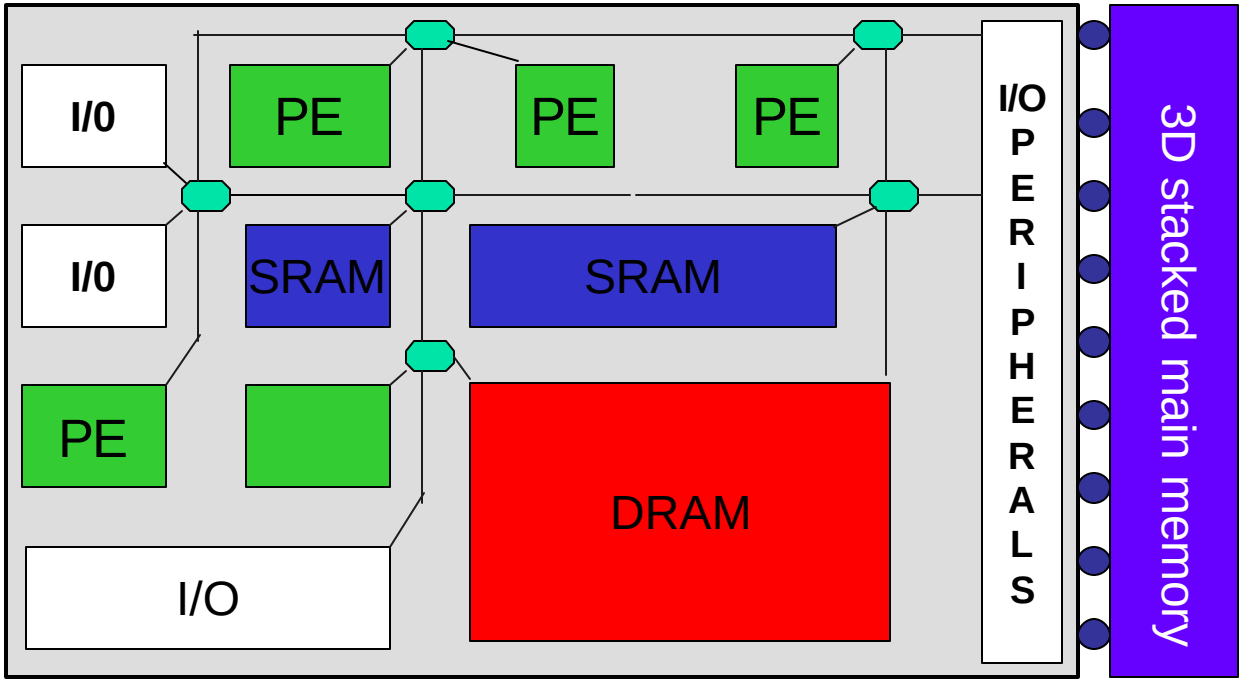


Picochip PC205 (Apr'06)

- 260MHz, 31GMAC/s, 160GIP/s
- 64KB I,D$, 128KB SRAM
- Less than 5 W, less than 1$/GMAC

Architecture Evolution

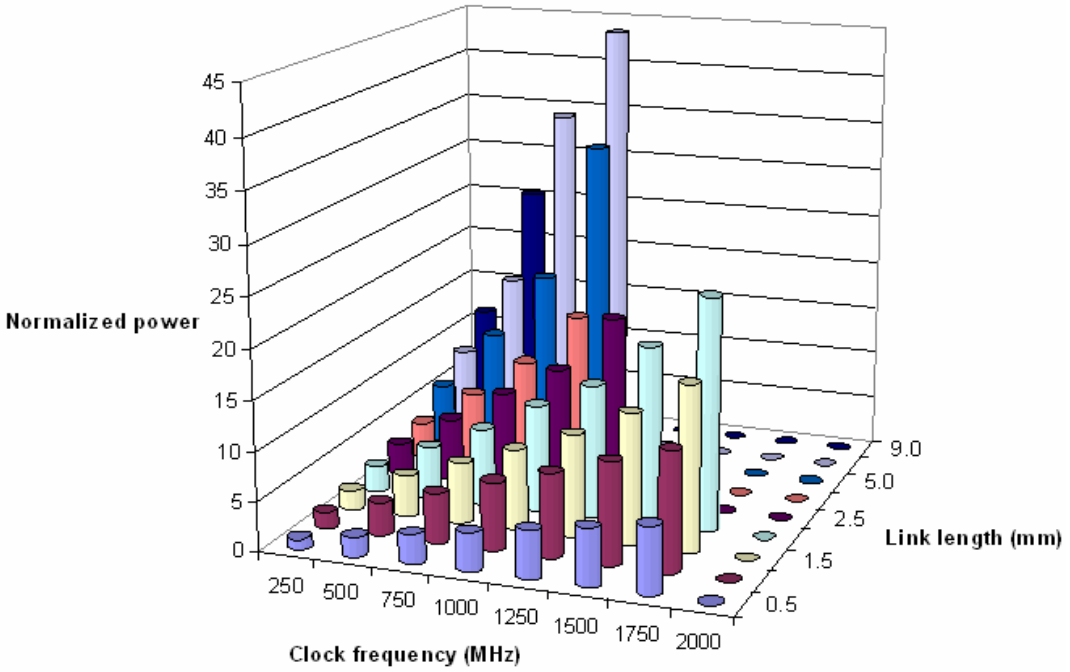




- Roadmap continues: 90→65→45 nm
- "Traditional" Bus-based SoCs fit in one tile !!
- Communication demand is staggering, but unevenly distributed, because of architectural heterogeneity

Interconnect Bottleneck

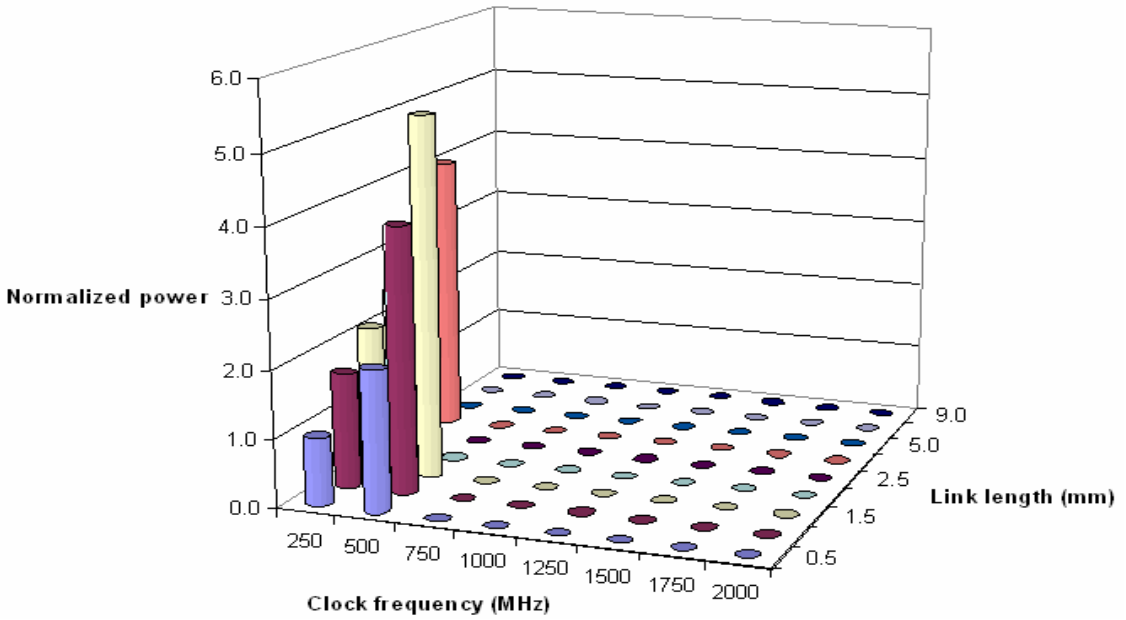
Power consumption
Unidirectional link
(38 bits+flow control)



- 65 nm low-power library
- low V_t library, high V_{DD} – power/perf tradeoff
- very high frequencies or very long links infeasible
- but even some feasible links burn up to 30 mW!!
  - heavy buffer insertion

Interconnect Bottleneck

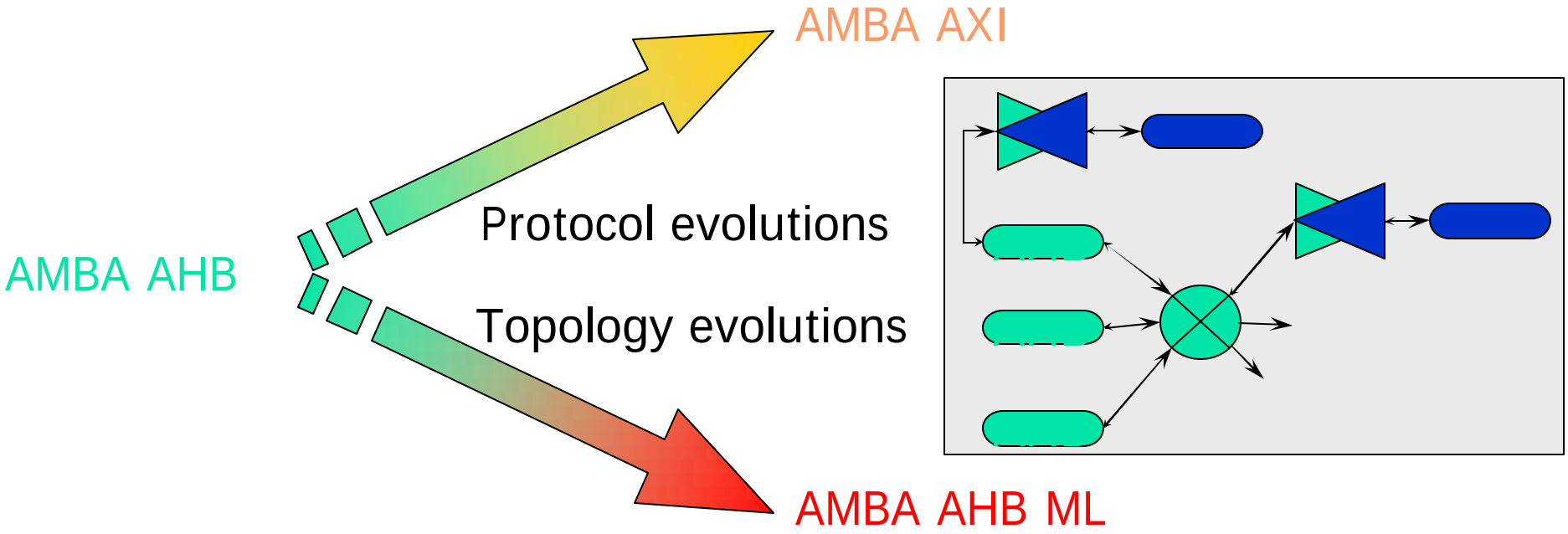
Power consumption
Unidirectional link
(38 bits+flow control)



- 65 nm low-power library
- High V_t library, low V_{DD} – absolute min power
- Even at 250 MHz, > 2 mm link length infeasible

Addressing Interconnect Issues

- High-end industrial solutions:
  - Evolutionary path from shared busses

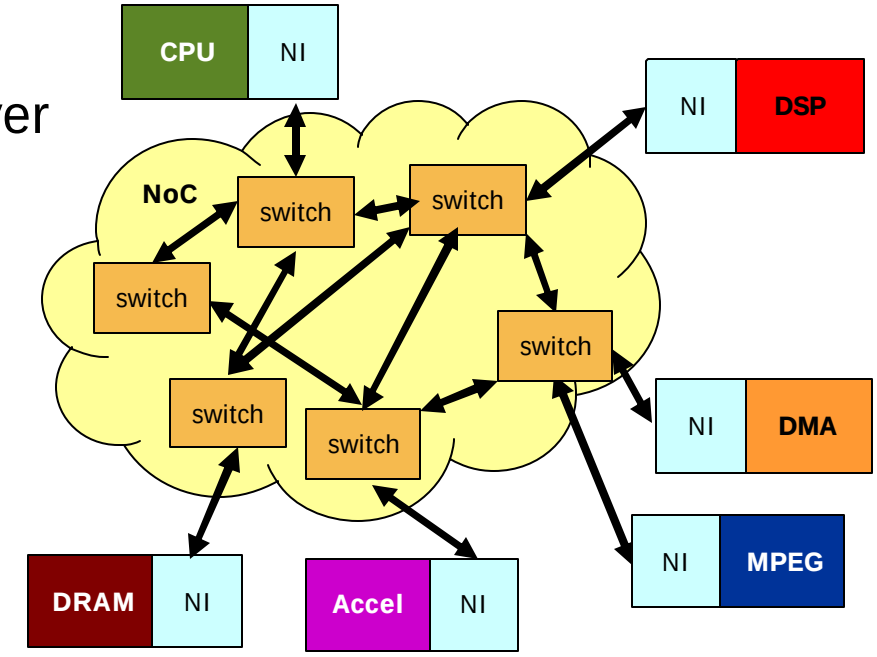


- Challenges
  - Complexity: how to analyze, verify “spaghetti interconnects”?
  - Scalability: bus is bandwidth-limited, Xbar is size-limited
  - Predictability: how to tie interconnects with floorplanning

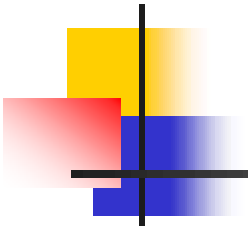
The Network-on-Chip Paradigm

The “power of NoCs”:

- Clean separation at session layer
  - Cores issue end-to-end transactions
  - Network deals with transport, network, link, physical
- Modularity at HW level: only 2 building blocks
  - Network interface
  - Switch (router)
- Physical design aware (floorplan global routing)



Scalability is supported from the ground up!

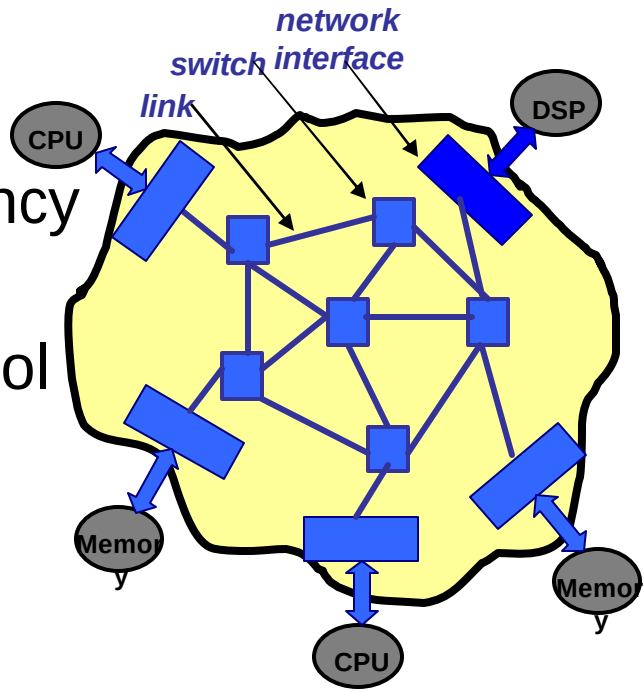


SoC and NoC Characteristics

- Typical applications targeted by SoCs
  - Complex
  - Highly heterogeneous (component specialization)
  - Communication intensive
- Tailor-made interconnects for applications
- NoCs are resource constrained:
  - Power, area constraints – low buffering available
- Large available wire bandwidth
  - But tapping it with modular, structured design is key

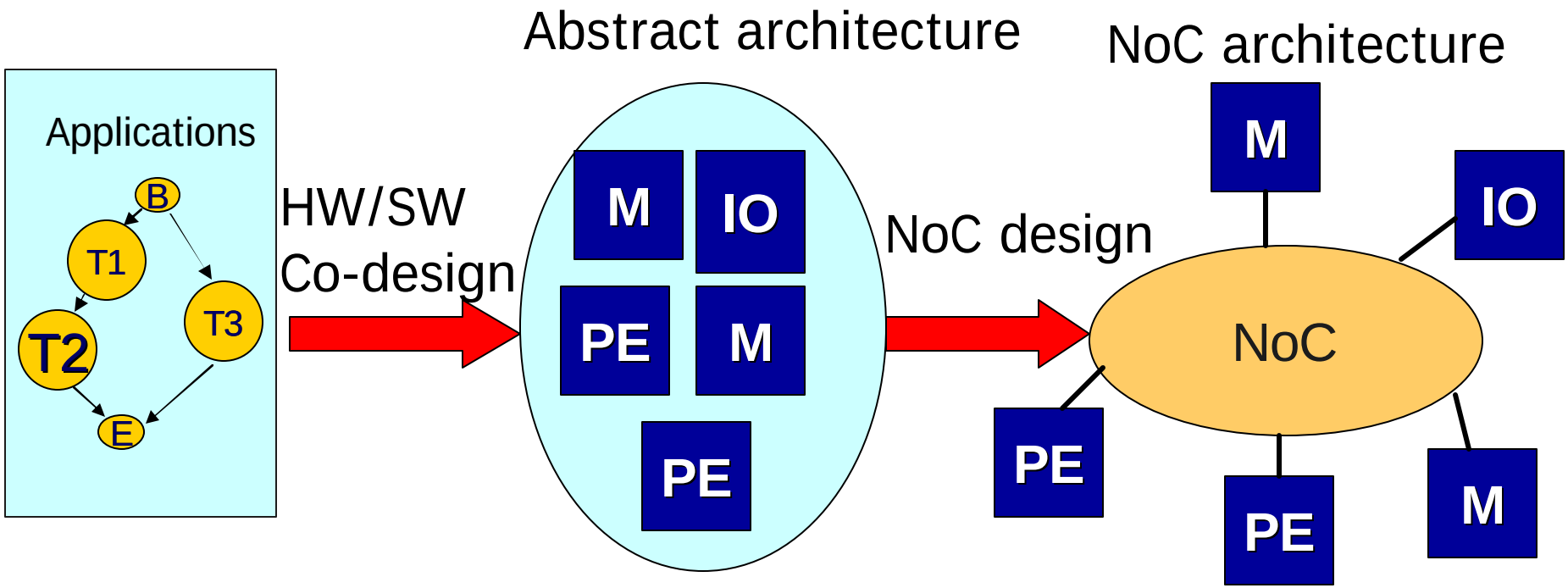
New design challenges

- From multiprocessor field
  - Assigning tasks to processors
  - Synchronization, consistency, coherency
- Networking
  - Network topology, routing, flow control
  - Quality of Service (QoS) needs
- VLSI
  - Floorplan in 2D, wire lengths
  - Power, area, performance



An integrated design approach is crucial ...

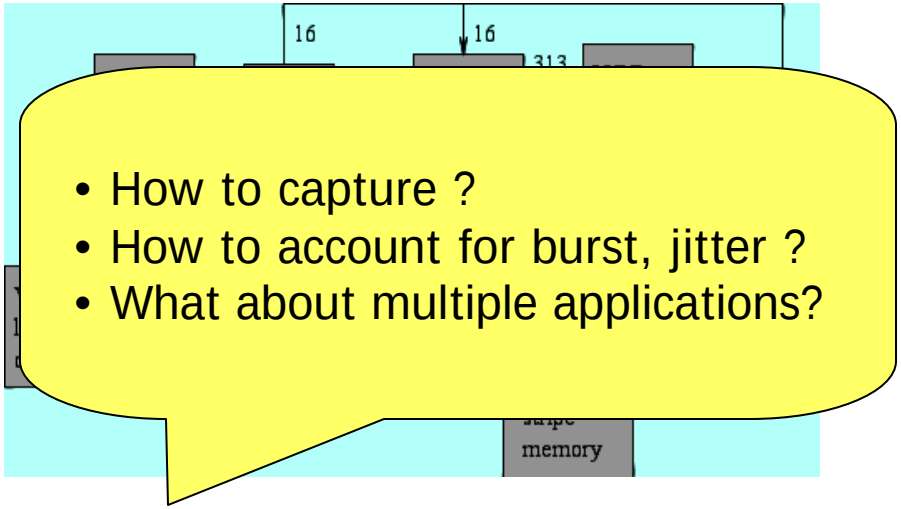
The Big Picture



Orthogonalize computation from communication

Why Design Automation ?

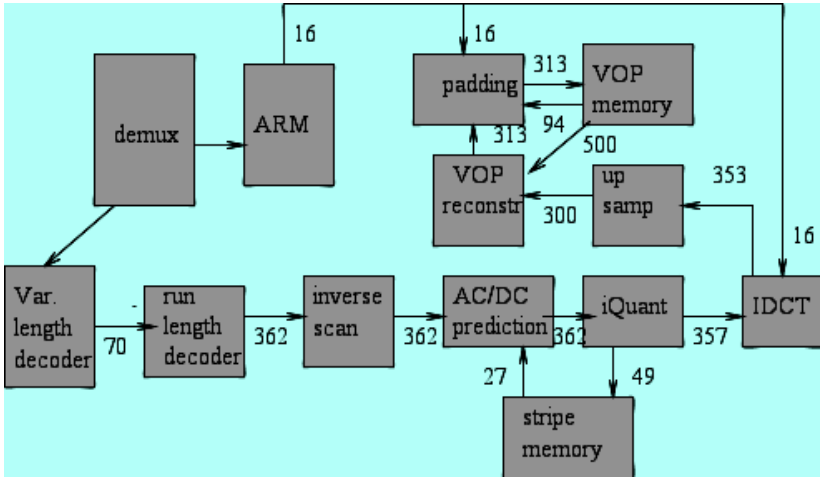
- Large design space, several steps



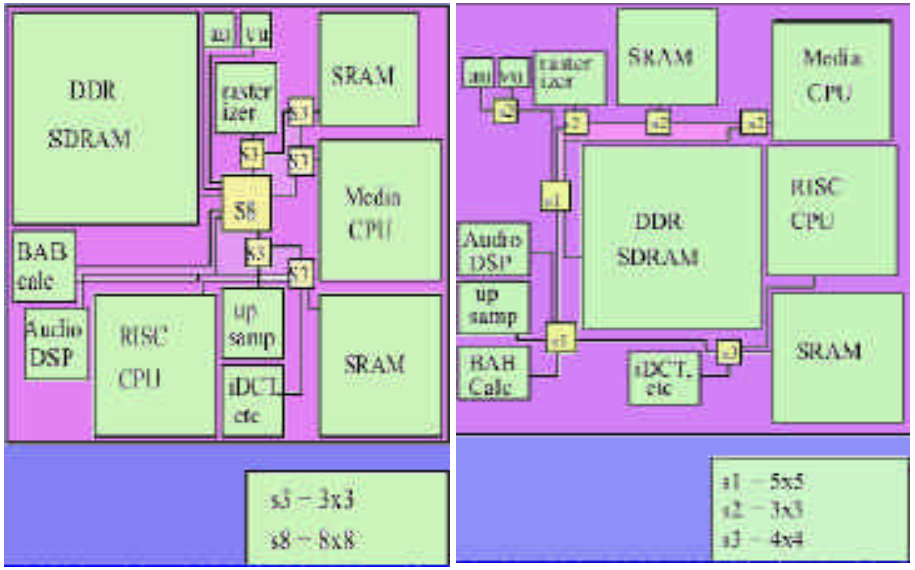
1. Capturing application traffic

Why Design Automation ?

- Large design space, several steps



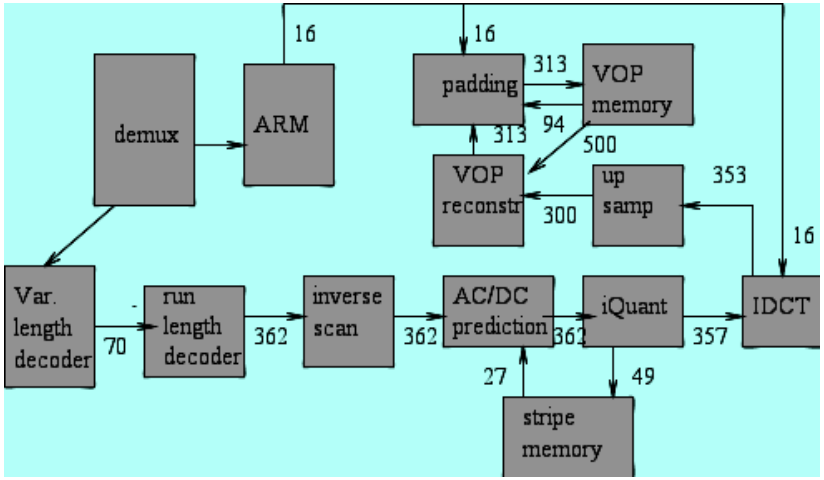
1. Capturing application traffic



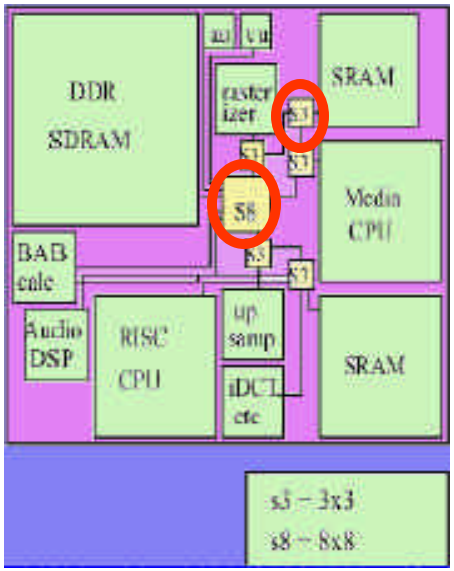
2. What topology ?
3. Mapping ?
4. Routes to use ?

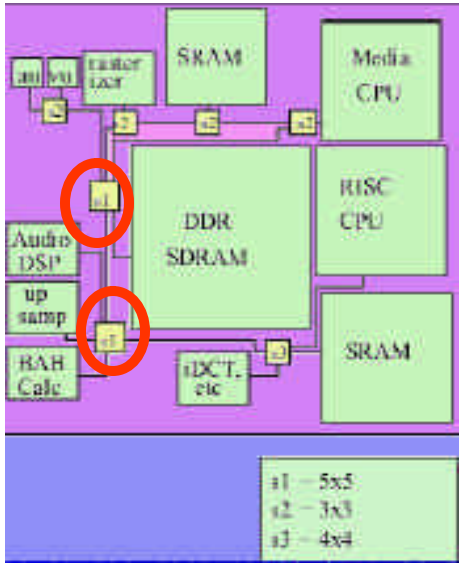
Why Design Automation ?

- Large design space, several steps



1. Capturing application traffic





-Resource constrained: power, area

-Large wire bandwidth - tapping it with modular design is key

2. What topology ?
3. Mapping ?
4. Routes to use ?

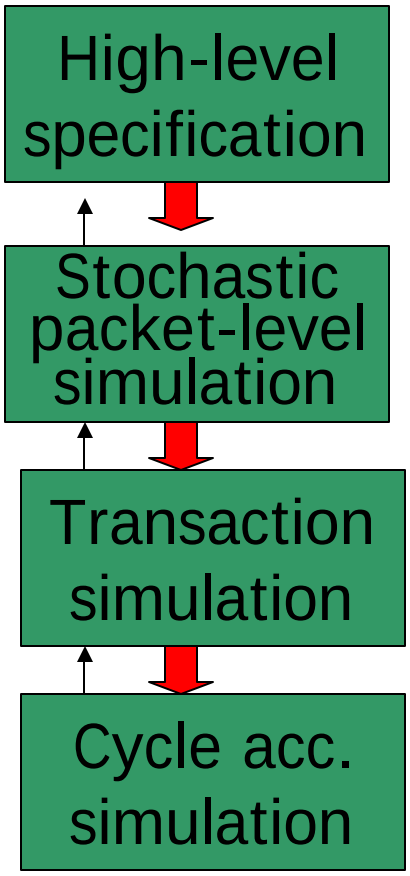
More Steps !

5. Tuning communication architecture parameters (link width, buffer sizes)
6. Verification for correctness, performance
7. Build simulation, synthesis, emulation models
8. Reliable operation under unreliable conditions

Should ensure design closure (fast time-to-market)

Automating and integrating the steps essential !

Layered Design Flow



Design phases	Models/effects	Key Issues
Topology design, mapping, routing, refine arch. parameters	Analytical models, static effects, large solution space	Accurate traffic modeling, performance, power modeling
Buffer sizing, arbitration policy, dynamic routing	Dynamic, fast C++ simulations, stochastic traffic	Traffic generator models, accurate network models
Further refine arch params, key topology changes	Dependencies in communication	Reflect cycle-accuracy, speed
Performance test, very few arch, topology changes	Completely accurate	Speed, FPGA emulation

Research Teams

Technion

Brazil

Princeton

KTH, Sweden

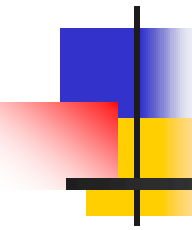
University of Bologna

Stanford

Tampere University of Technology

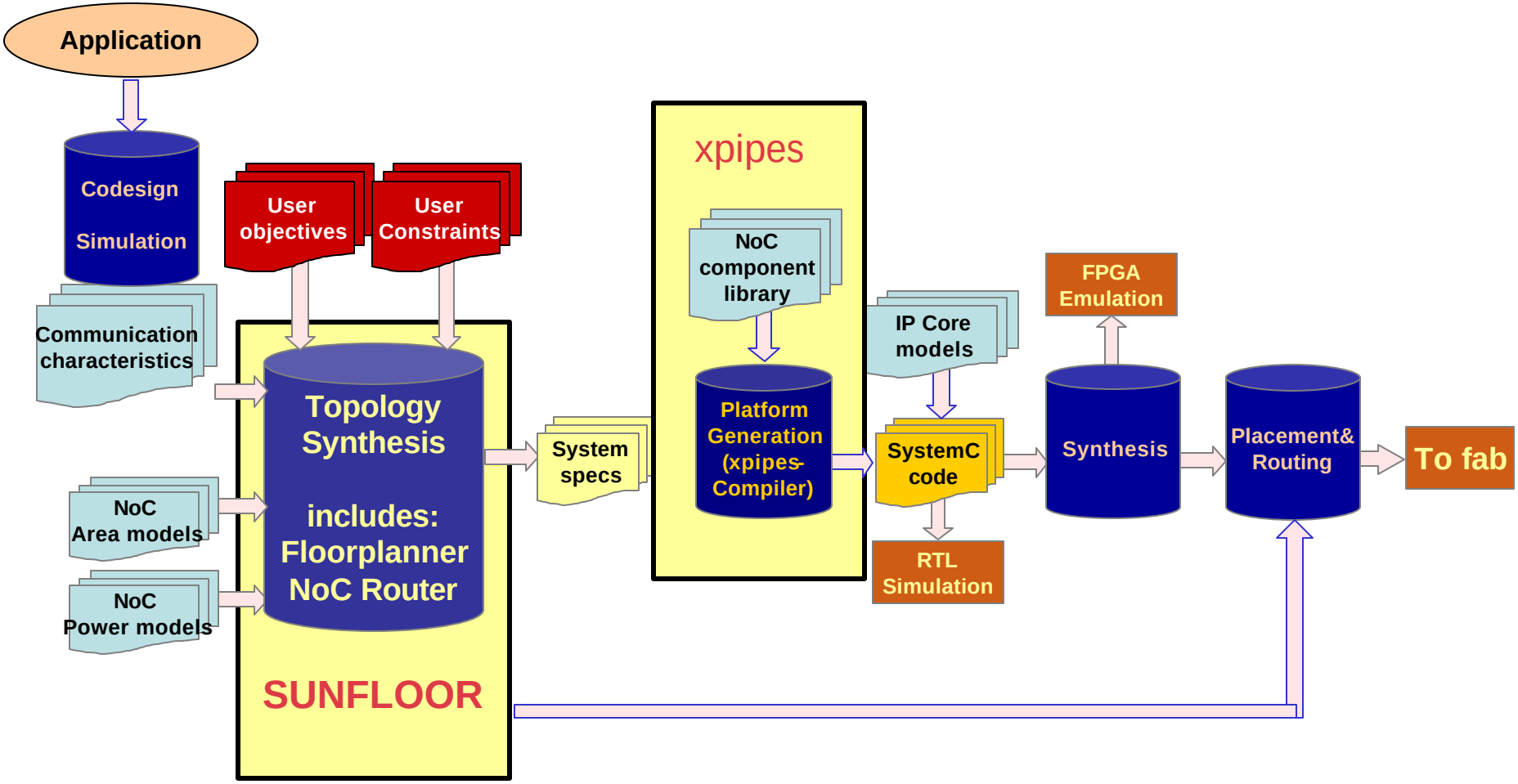
University of Cagliari

All omissions are purely accidental ...



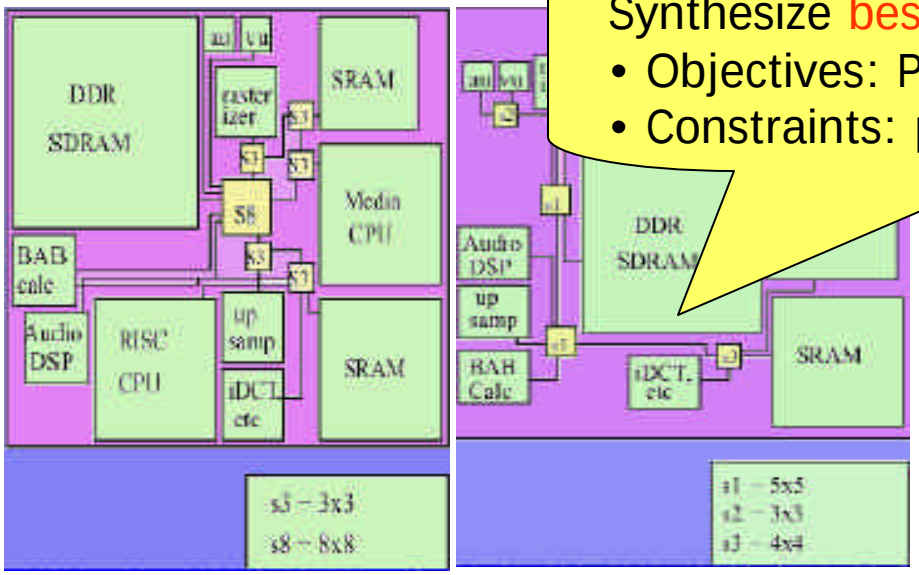
SunFloor Design Flow

SunFloor Design Flow



Front-End Design

- Design application-specific custom topologies



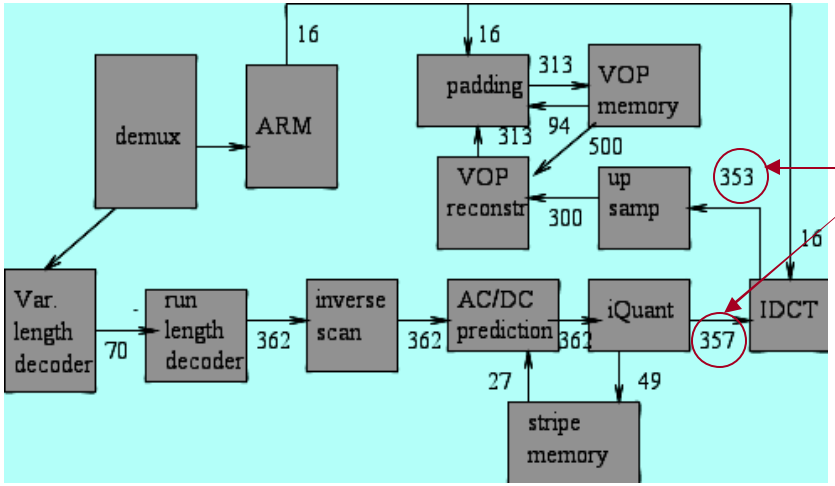
Synthesize best topology for application
- Objectives: Power, performance (hop delay)
- Constraints: performance, power, bandwidth

- Tune NoC frequency: match needs
- Design deadlock-free network
- Consider timing constraints early in design cycle
- Use accurate floorplan information

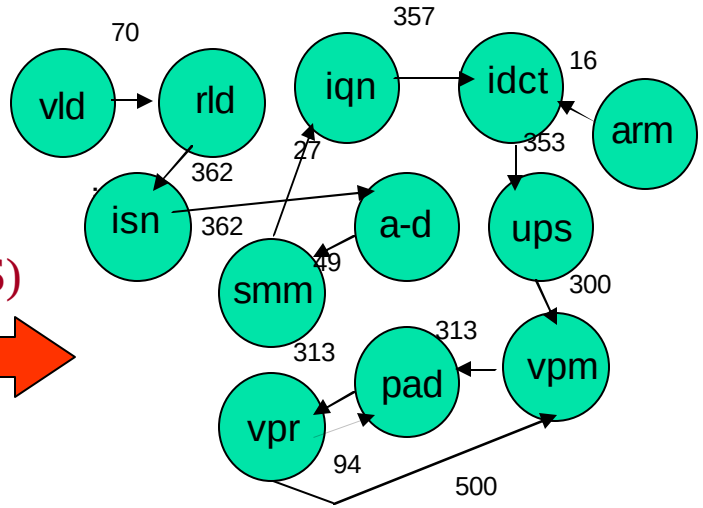
Achieves design closure, bridging design gaps across different steps

Input Models

- Traffic Models



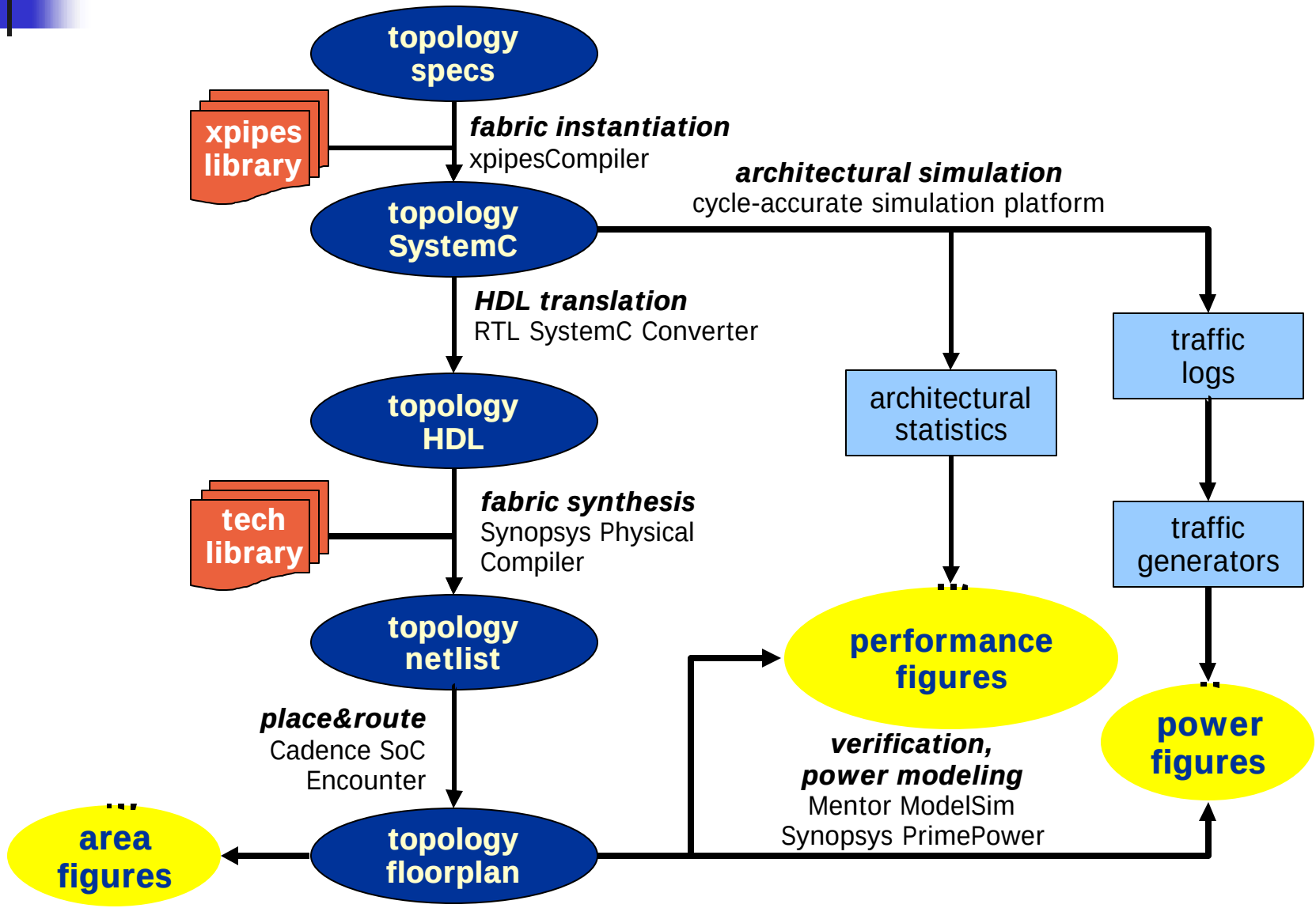
BW (MB/S)

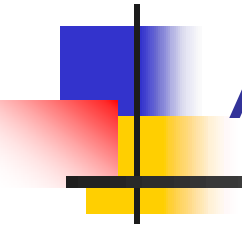


Core graph

- Consider bursty traffic, criticality of streams
- Obtained from initial simulations, application knowledge
- Hardware monitors to obtain traffic characteristics

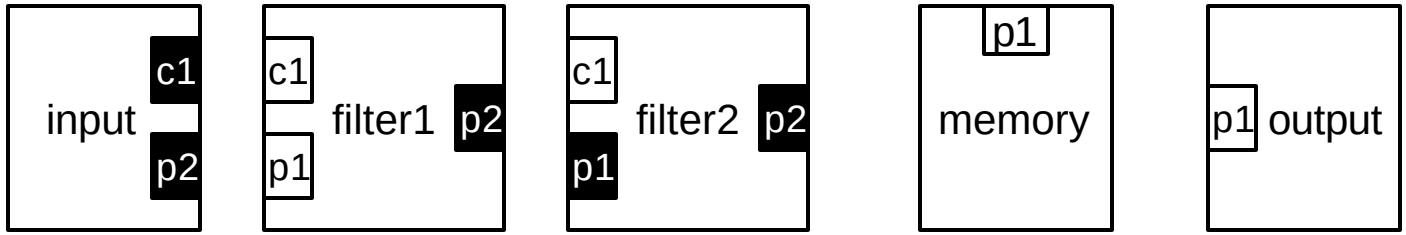
Back-End Flow





Æthereal Design Flow

Architecture Specification



architecture.xml

H:\projects\flow\examples\small\small_architecture.xml - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media

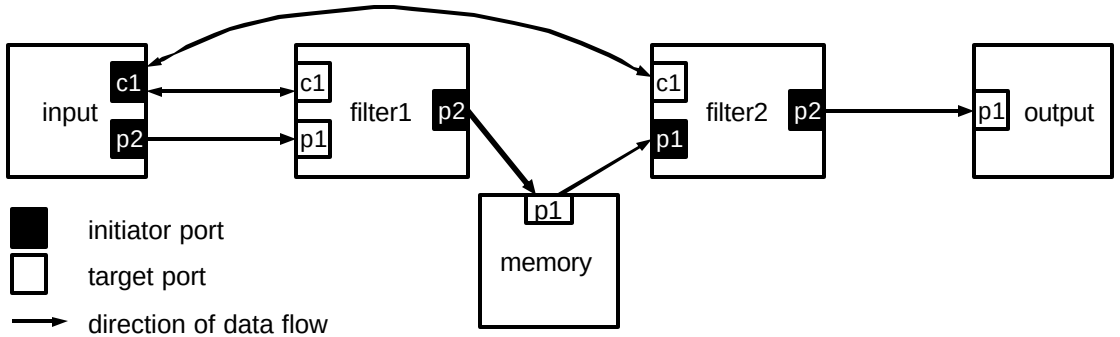
Address H:\projects\flow\examples\small\small_architecture.xml Go Links

```
- <architecture id="small">
  - <IP id="input">
      <initiator id="c1" protocol="DTL MMIO" width="32bit" />
      <initiator id="p2" protocol="DTL MMBD" width="32bit"
        blocksize="32" />
    </IP>
  + <IP id="filter1">
  + <IP id="filter2">
  + <IP id="memory">
  + <IP id="output">
```

Local intranet

[Kees Goossens, NXP]

Application specification



communication.xml

Microsoft Excel - small.xls

File Edit View Insert Format Tools Data Window Help

Arial 10

F17 =

	A	B	C	D	E	F	G	H	I
1									
2	Initiator port	Target port	Read			Write			QoS (GT/BE)
3			Bandwidth (MBytes/sec)	BurstSize (Bytes)	Latency (nano sec)	Bandwidth (MBytes/sec)	BurstSize (Bytes)	Latency (nano sec)	
4	input_c1	filter1_c1	40	32	100	24	32	100	BE
5	input_c1	filter2_c1	50	32	100	70	32	100	BE
6	input_p2	filter1_p1	0	0	0	240	32	0	GT
7	filter1_p2	memory_p1	500	32	0	0	0	0	GT
8	filter2_p1	memory_p1	500	32	0	0	0	0	BE
9	filter2_p2	output_p1	0	0	0	240	32	0	
10									
11									

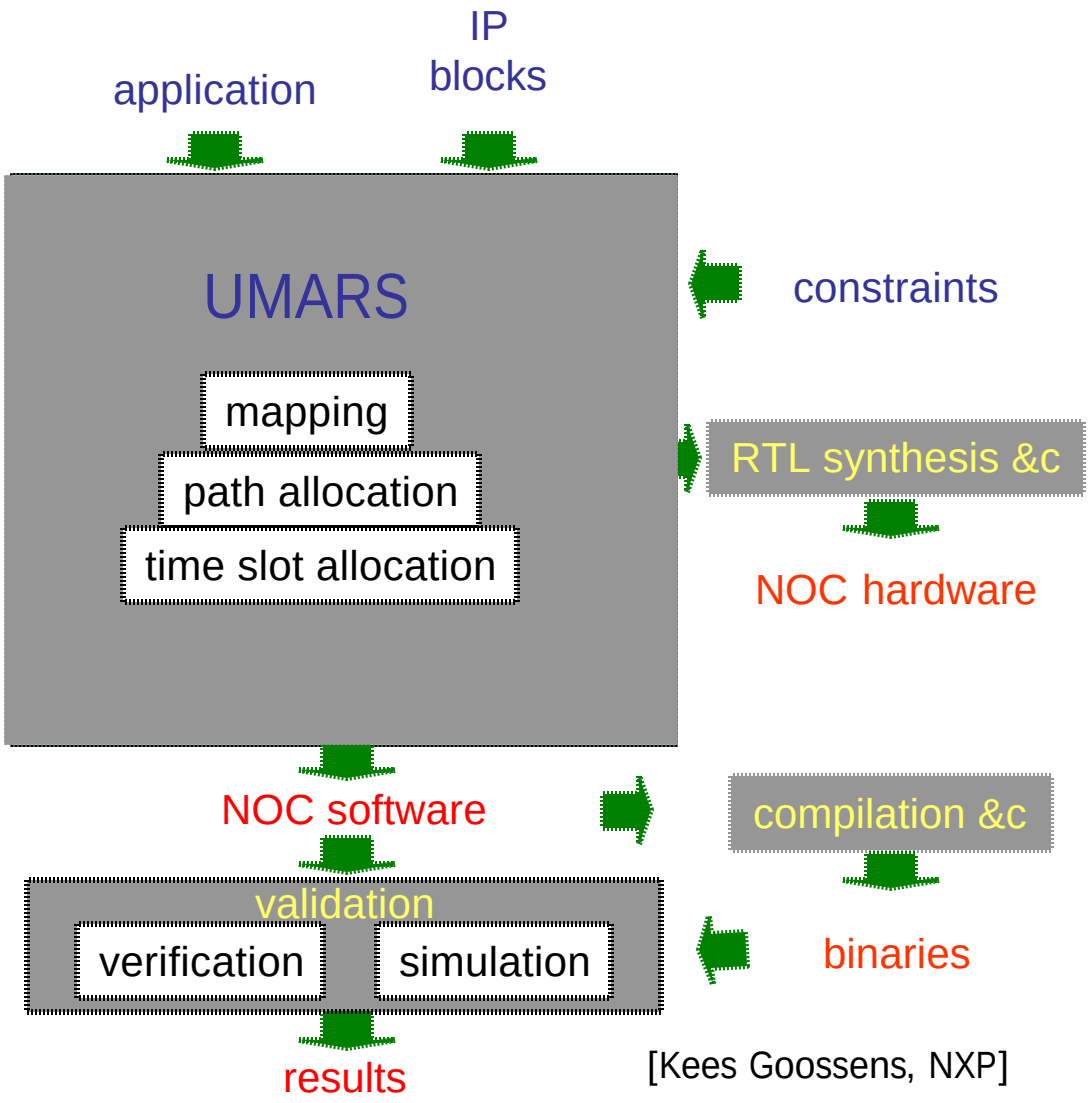
one complex more-complex

Ready NUM

[Kees Goossens, NXP]

NOC design flow

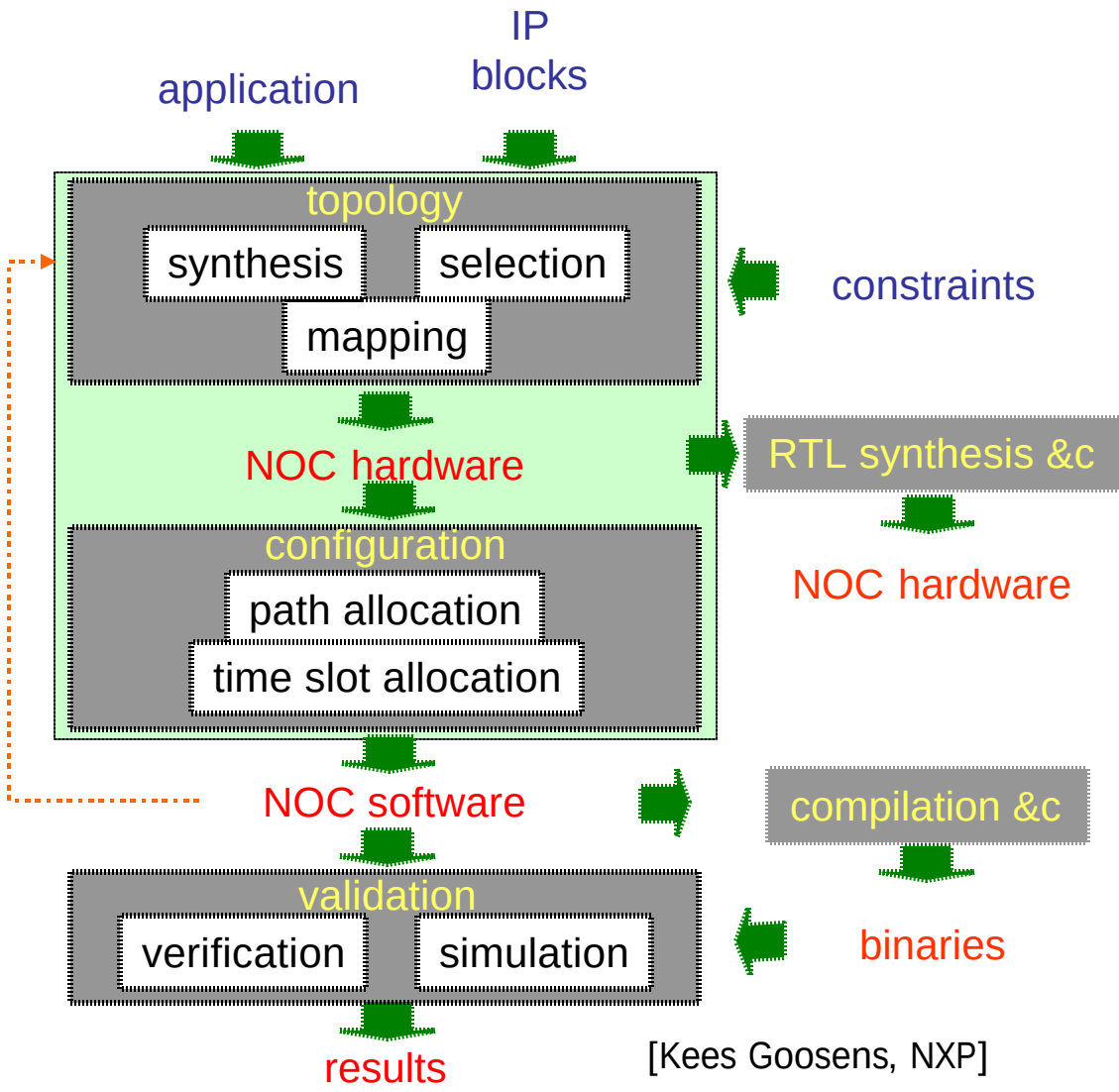
- Split large optimization problem in smaller pieces



[Kees Goossens, NXP]

NOC design flow

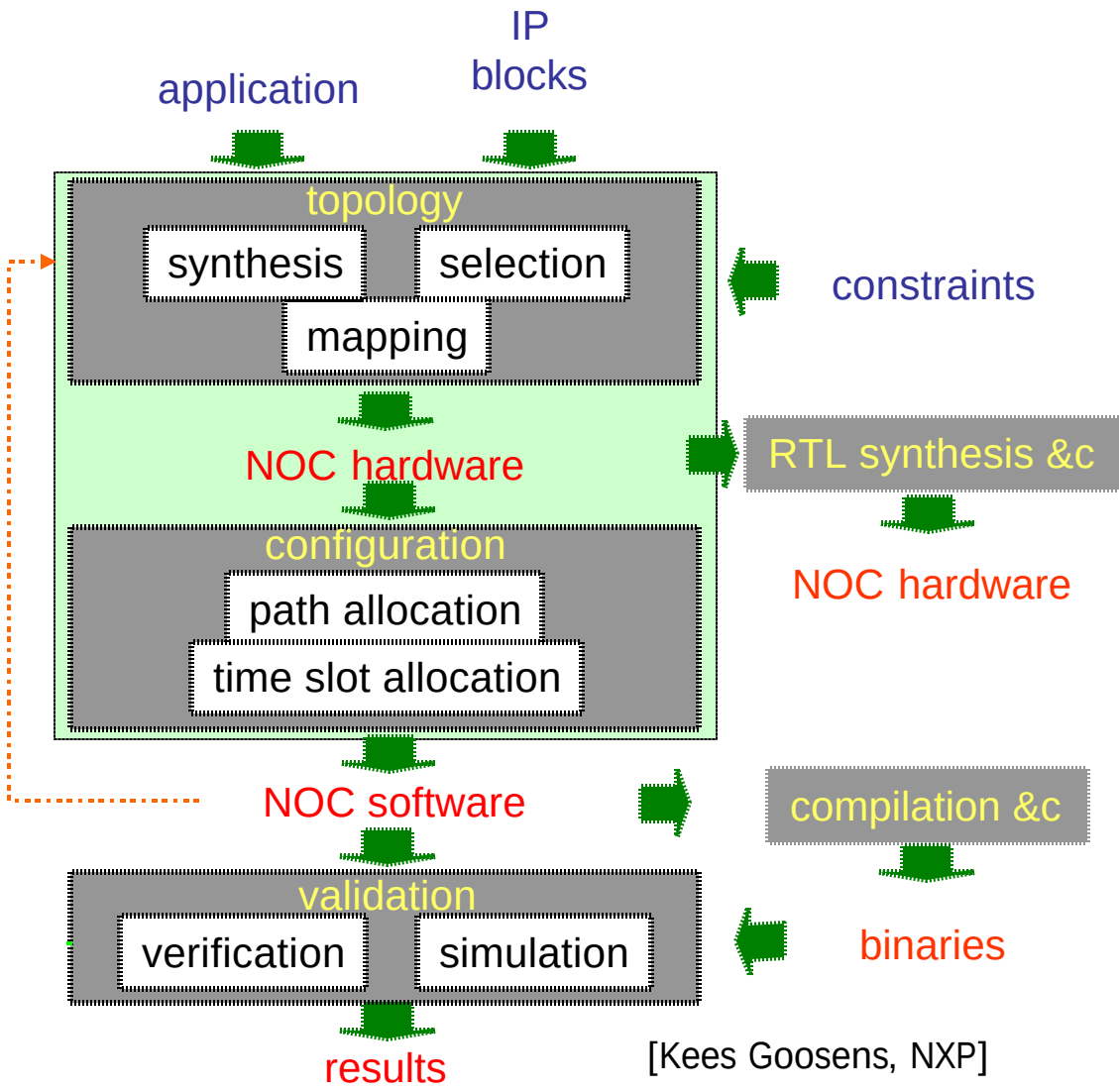
- Split large optimisation problem in smaller pieces
  - may fail (feedback)



[Kees Goosens, NXP]

NOC design flow

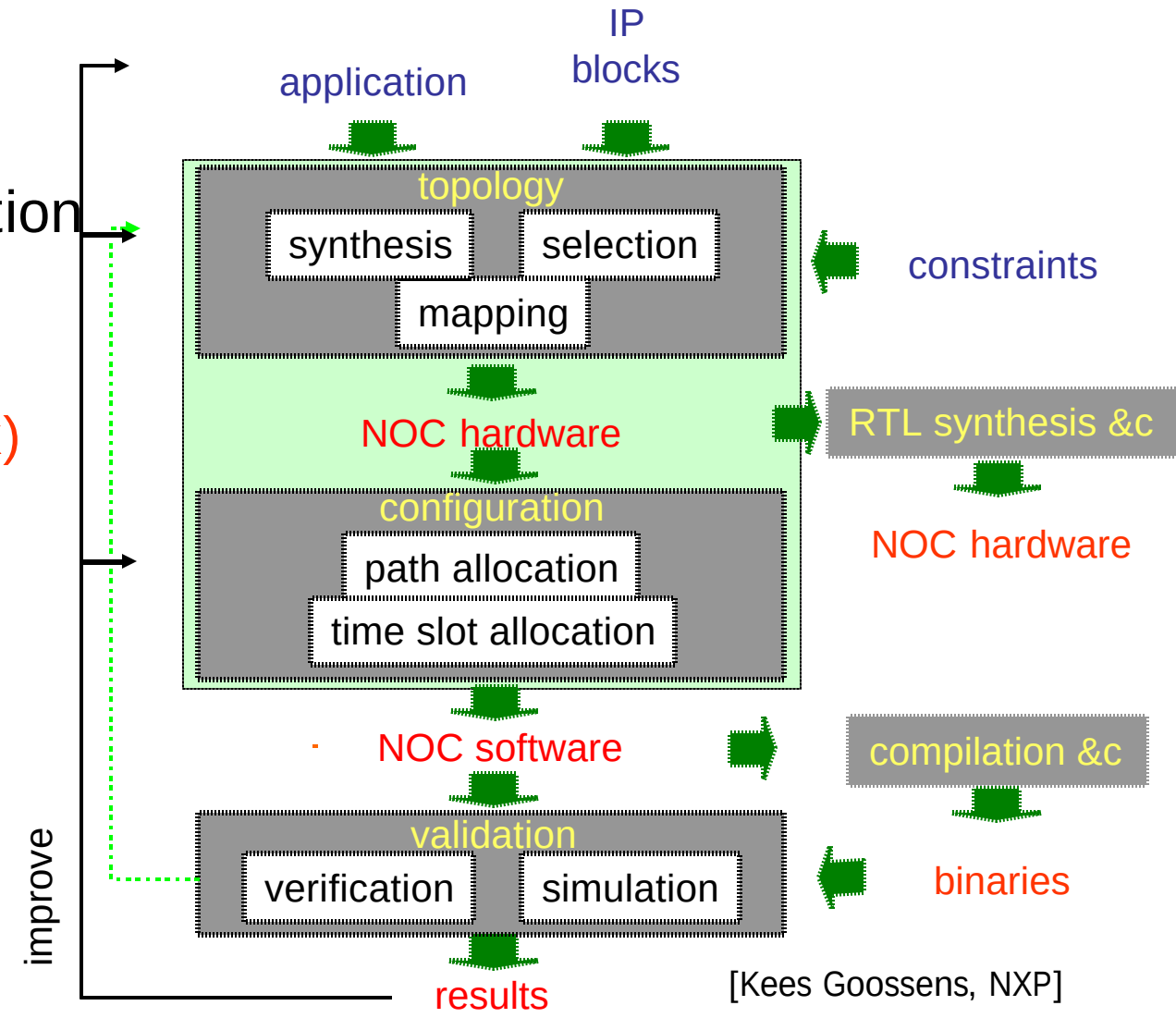
- Split large optimisation problem in smaller pieces
  - may fail (feedback)
  - back annotation



[Kees Goosens, NXP]

NOC design flow

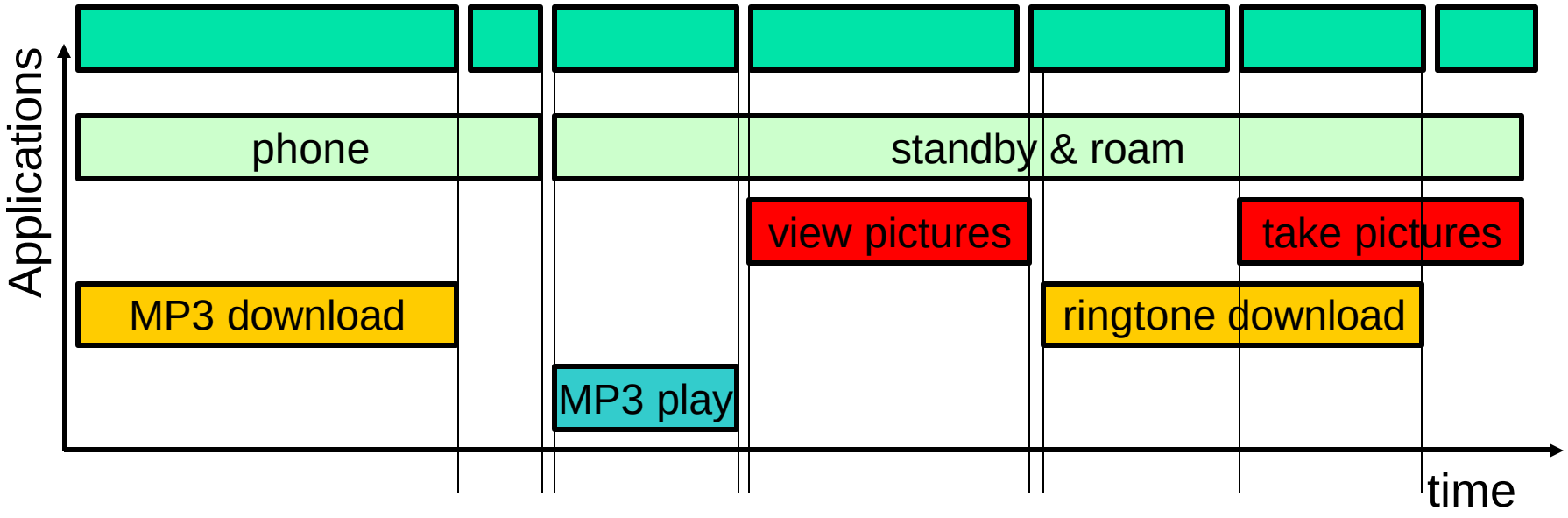
- Split large optimisation problem in smaller pieces
  - may fail (feedback)
  - back annotation



[Kees Goossens, NXP]

UMARS: Multiple applications

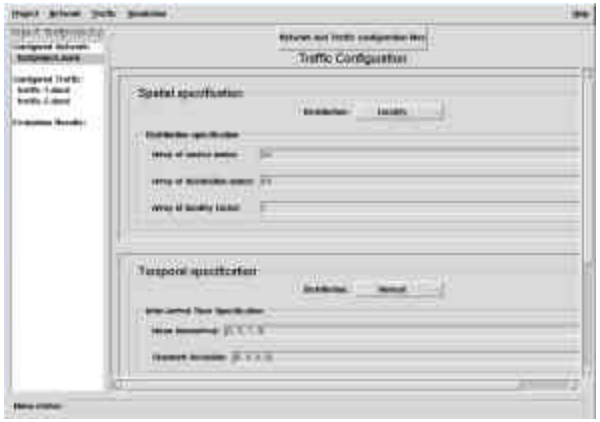
- SoCs typically support multiple applications
- Applications can run in parallel: compound modes
- UMARS supports multiple applications
  - Supports NoC reconfiguration across compound modes



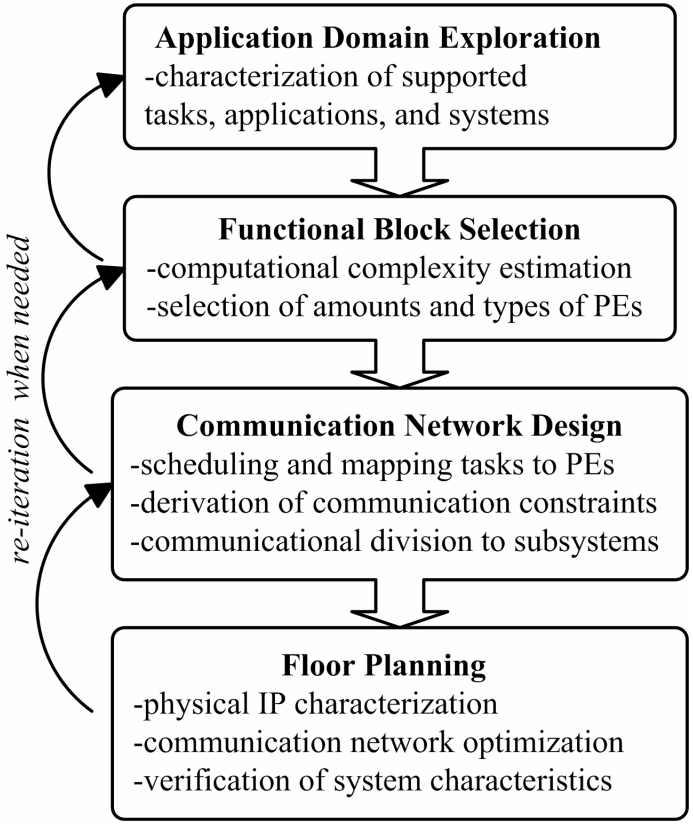
[Kees Goossens, NXP]

Several NoC CAD efforts

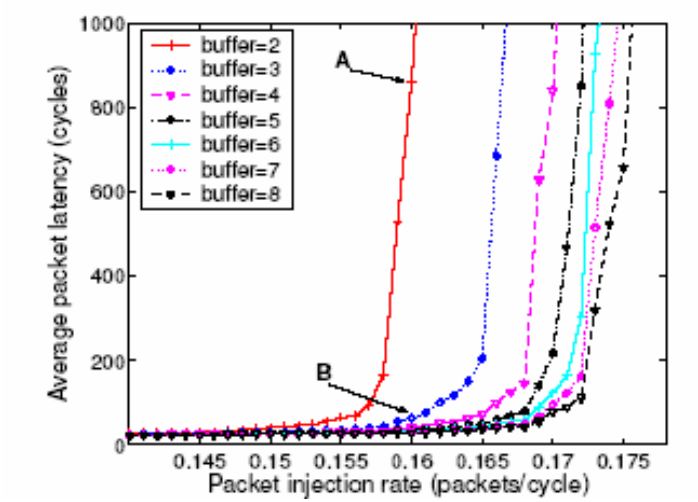
Nostrum simulation environment

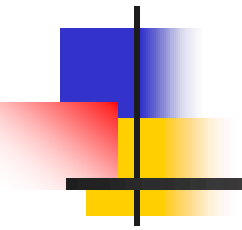


OEDIPUS design system [Ahonen]



NoC buffering with queueing theory [Hu]





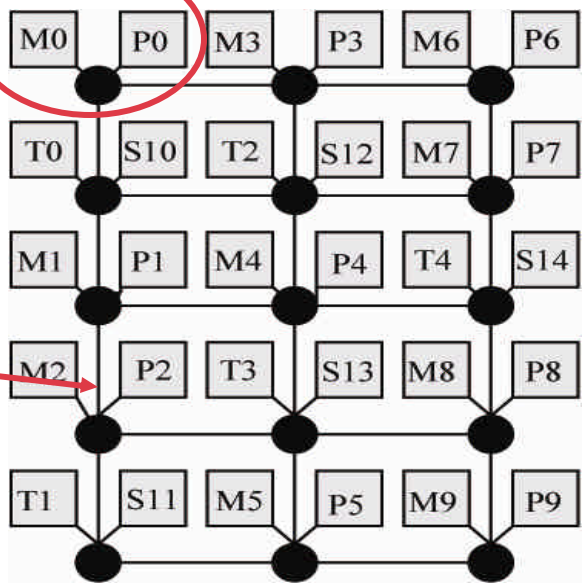
Case Study 1: Custom Vs Regular NoCs

SUNFLOOR vs Manual design

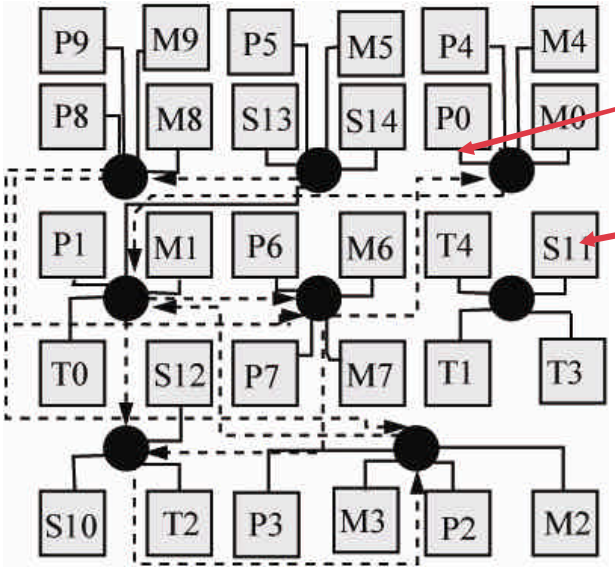
On the 30-core multimedia benchmark

Processor-memory cluster

Bi-directional links



Hand-mapped topology



Uni-directional links

Bi-directional links

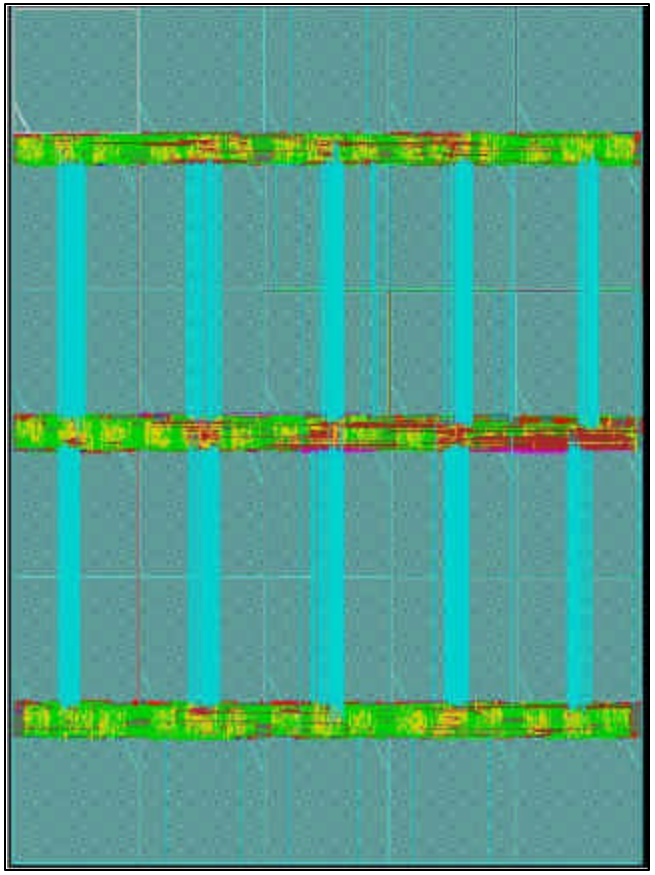
SUNFLOOR custom topology

P-processors, M-private memories,

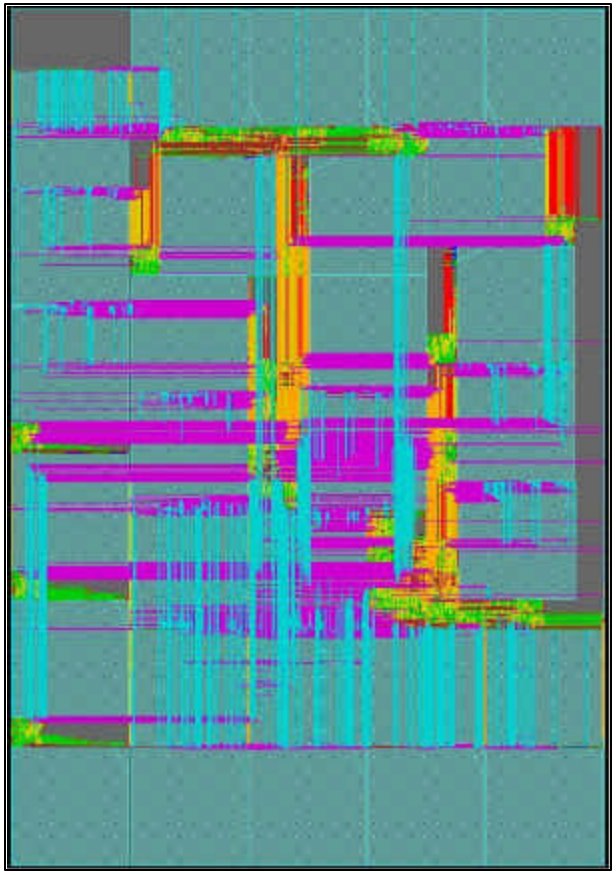
T-traffic generators, S-shared slaves

Design Layouts

From Cadence SoC Encounter



Hand-design (custom mesh)



SUNFLOOR Design

SUNFLOOR vs Hand-Mapped

constraint

Hand-mapped design:

- Topology: 5x3 mesh (15 switches)
- Operating frequency: 885 MHz (post-layout)
- Power consumption: 368 mW
- Floorplan area: 35.4 mm^2
- Design time: weeks
- 0.13 µm technology

SunFloor:

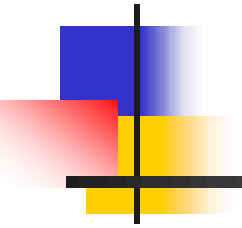
- Topology: custom (8 switches)
- Operating frequency: 885 MHz (post-layout)
- Power consumption: 277 mW **(-25%)**
- Cell area: 37 mm^2 **(+4%)**
- Design time: **4 hours design to layout**
- 0.13 µm technology

Benchmark execution time comply with application requirements and are even 10% better on SunFloor topology.

Custom Vs Regular Topologies

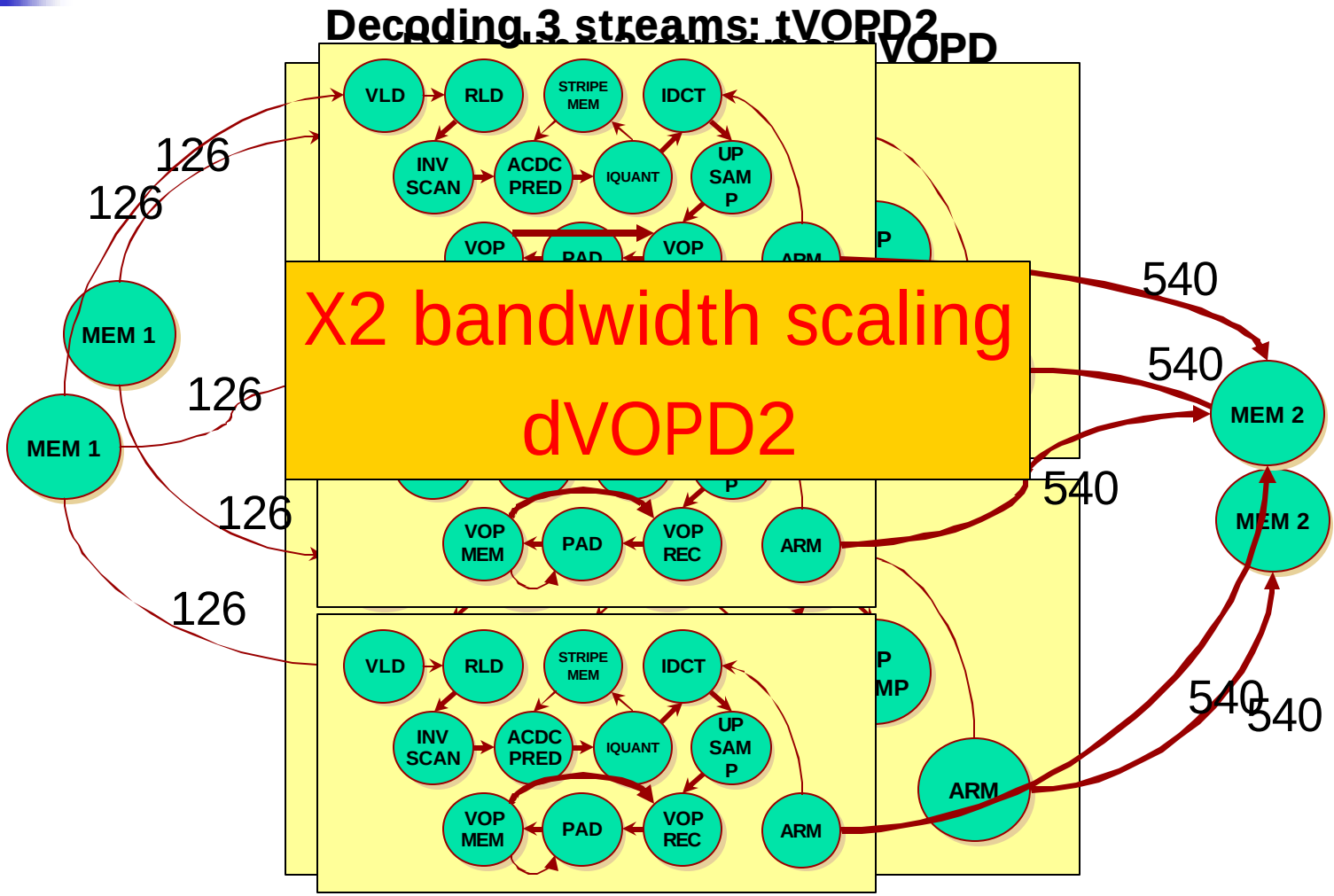
Application	Topology	Power(mW)	Avg. nr. hops
VPROC (42 cores)	Custom	79.64	1.67
	Mesh	301.8	2.58
	Opt-mesh	136.1	2.58
MPEG4 (12 cores)	Custom	27.24	1.50
	Mesh	96.82	2.17
	Opt-mesh	60.97	2.17
VOPD (12 cores)	Custom	30.00	1.33
	Mesh	95.94	2.00
	Opt-mesh	46.48	2.00
MWD (12 cores)	Custom	20.53	1.15
	Mesh	90.17	2.00
	Opt-mesh	38.60	2.00

- On average, SunFloor custom topologies:
  - **2.75x** less power consumption
  - **1.55x** less hop delay
- Despite large design space, maximum run time of few hours for VPROC



Case Study 2: Technology Scaling Effects

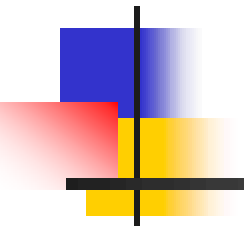
Effect of Technology Scaling



Network Synthesis Results

	Library	Frequency	Switch Count	Largest Switch	Total NoC Power	Avg. head flit latency
dVOPD	90nm HP					les
	90nm LP					
	65nm HP					les
	65nm LP	-	-	-	-	-
dVOPD2	65nm HP	800 MHz	6	7x6	129.36 mW	4.24 cycles [3,7]
tVOPD2	65nm HP	800 MHz	10	7x7	196.40 mW	4.35 cycles [3,9]

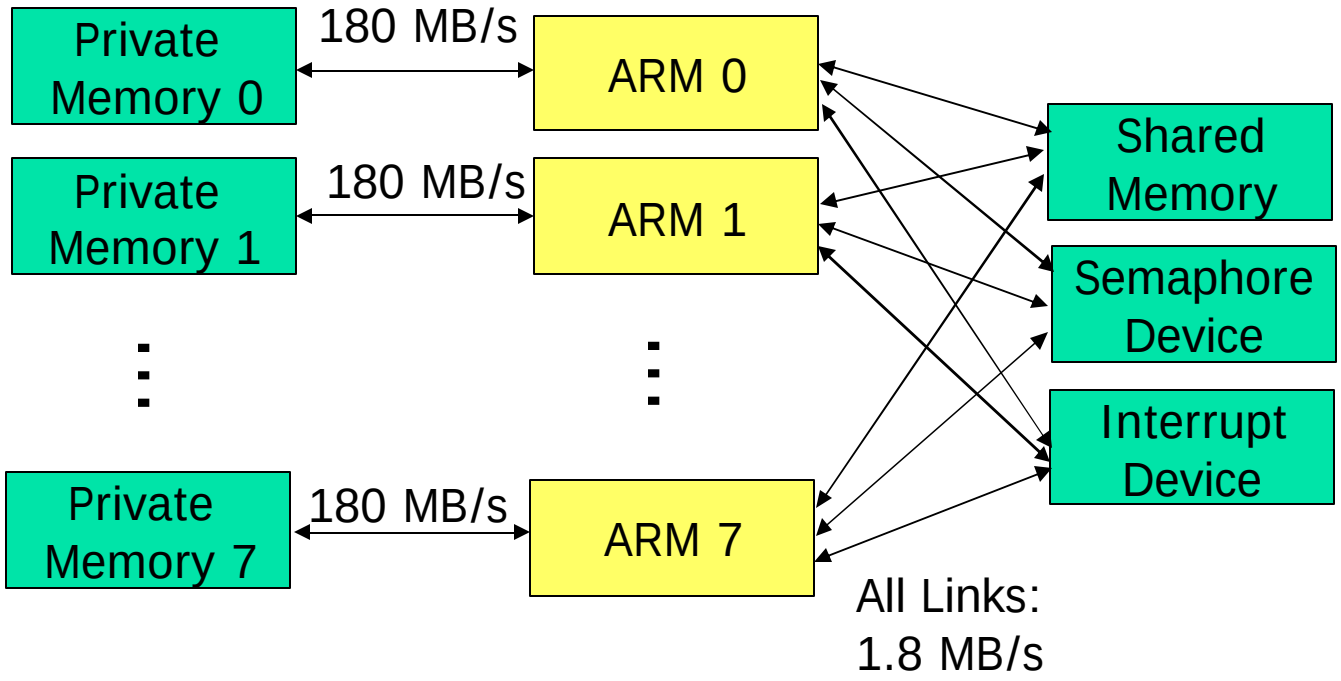
- Observations:
  - Lower power in 65nm for same design
  - 65 nm supports 2x BW, at lower power!
  - NoC for a big design (38 cores) operates at 800 MHz
  - With increasing app BW or number of cores, more switches needed (due to freq limit of switches)



Case Study 3:
NoCs for low power applications ?

Parallel Encryption Engine

- 18 cores

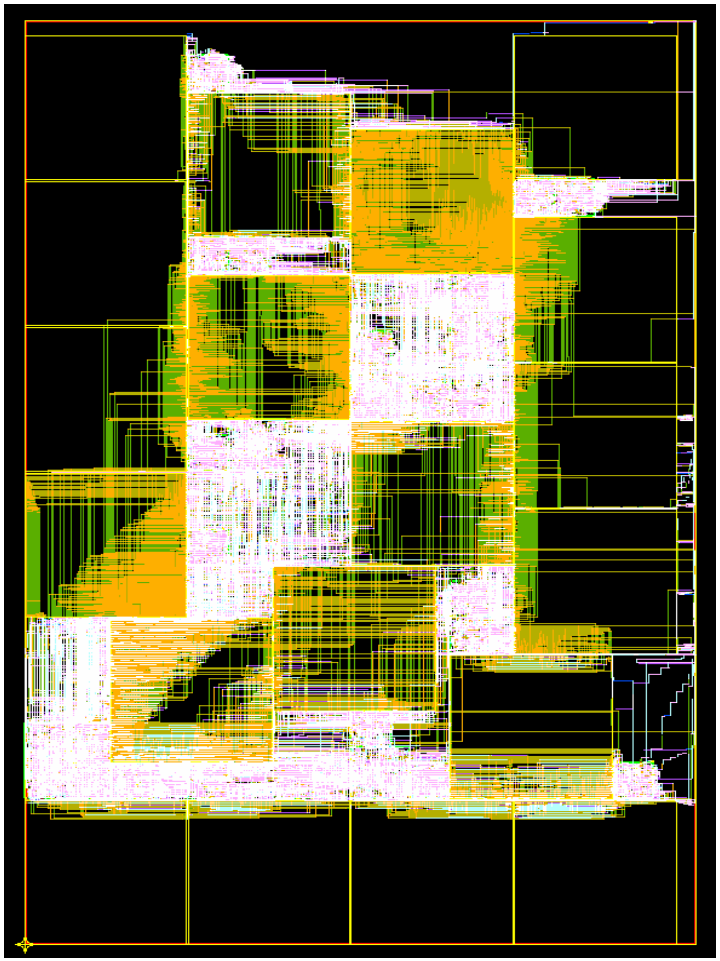


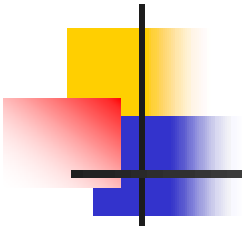
Low Bandwidth & Power Application

Library	Frequency	Switch Count	Largest Switch	Total NoC Power	Avg. head flit latency
90nm HP	50 MHz	2	11x11	10.4 mW	3.94 cycles [3,5]
90nm LP	50 MHz	2	11x11	4.1 mW	3.94 cycles [3,5]
65nm HP	50 MHz	2	11x11	4.72 mW	3.94 cycles [3,5]
65nm LP	50 MHz	5	9x9	3.1 mW	4.38 cycles [3,7]

Energy efficiency: 2.2Gbs/mW→ 2.5x better than high-perf NoC

Custom Topology Layout





Conclusions

- Design flows and CAD tools are critical for NoCs
  - Layered design flow
    - Tackle problems from several levels
  - Several key steps
    - Traffic analysis, mapping, topology design, routing,...
  - Integrated approach is critical
    - Interact with existing back-end tools
- Fertile ground for more R&D work:
  - Run-time configurability
  - Robustness w.r.t. to static/dynamic variations, errors
  - Tackle floorplan and layout issues